



Control of a Buoyancy-Based Pilot Underwater Lifting Body

Finn Haugen

Telemark University College, Kjoelnes Ring 56, N-3918 Porsgrunn, Norway. E-mail: finn.haugen@hit.no

Abstract

This paper is about position control of a specific small-scale pilot underwater lifting body where the lifting force stems from buoyancy adjusted with an air pocket in the lifting body. A mathematical model is developed to get a basis for a simulator which is used for testing and for designing the control system, including tuning controller parameters. A number of different position controller solutions were tried both on a simulator and on the physical system. Successful control on both the simulator and the physical system was obtained with cascade control based on feedback from measured position and height of the air pocket in the lifting body. The primary and the secondary controllers of the cascade control system were tuned using Skogestad's model-based PID tuning rules. Feedforward from estimated load force was implemented in combination with the cascade control system, giving a substantial improvement of the position control system, both with varying position reference and varying disturbance (load mass).

Keywords: Underwater, buoyancy, air lift, position control, cascade control, feedforward control, disturbance, estimation, Skogestad model-based controller tuning

1 Introduction

Underwater lifting operations is a common task in e.g. the offshore oil and gas production industry. This paper is about control of a small-scale physical pilot underwater lifting system.¹ The lifting principle is adjustment of the buoyancy by controlling the amount of air in the air pocket of the lifting body. Buoyancy can provide a large lifting force with little energy, but it requires a control system. A position control system is designed and implemented to keep the body with load at a reference position. A mathematical model is developed to get a basis for a simulator which is used for testing and for designing the control system, including tuning controller parameters.

Fossen (1994) and others, describe position control of underwater vehicles, but there is not much research reported about stabilizing underwater bodies using buoy-

ancy. Several control functions were tried both on a simulator and on the physical system. The only method which worked well on the physical system was standard cascade control with positional control as the primary (master) control loop and air mass in the lifting body as the secondary (slave) control loop. The selected control strategy has certain similarities with the control structure described in a US patent by Otterblad and Doovertie (1985) where the inner loop is based on a measurement of the lifting force.² The simulator and the control system are implemented in a LabVIEW program running with cycle time 0.02 sec on a PC.

The paper is organized as follows: The system is described in Section 2. A mathematical model is derived in Section 3. This model is the basis of a simulator of the system, and it is also used for design and tuning of the controllers and the observer (estimator). Con-

¹This project was initiated and funded by the company Miko Marine, Oslo, Norway.

²Actually, in the patent the secondary loop is based on the time-derivative of the force measurement.

control system design, including state observer design, is described in Section 4. Experimental results are presented in Section 5. Conclusions are given in Section 6.

2 System Description

Fig. 1 shows schematically the lifting body, including the position control system. Fig. 2 (left) shows a photo of the lifting body inside a water tank used for testing the position control system. Fig. 2 (right) shows the actuator which is a pneumatic control valve which adjusts the air flow into the air pocket of the lifting body. The air flows out of the air pocket via a valve with fixed (manually adjustable) opening.

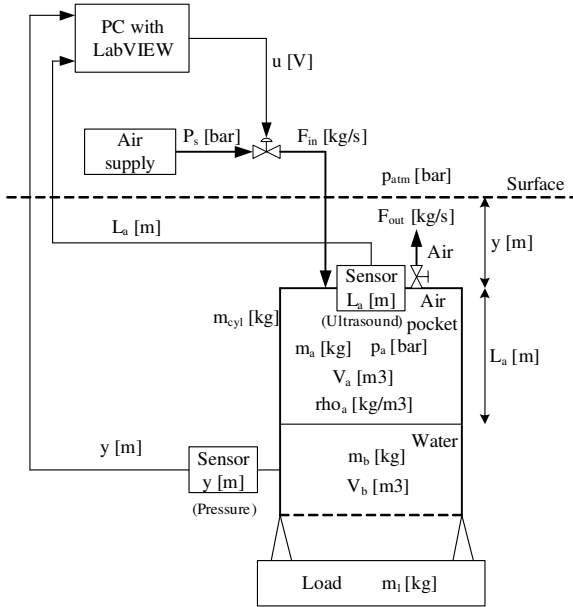


Figure 1: *The lifting body with load.*

The lifting body position y is measured with a pressure sensor. The position is calculated from the pressure measurement, since the position is proportional to the hydrostatic pressure. The height of the air pocket in the lifting body is measured with an ultrasound level sensor.³

It is certainly an open question whether the configuration shown in Fig. 1 represents optimal design. Alternative configurations are controlling the outflow, and controlling both inflow and outflow simultaneously with split-range control. However, it was not a purpose of the present project to find the optimal configuration

³It was important that the cap of the lifting body where the sensor was mounted was isolated (we used foam plastic), to avoid disturbing sound reflections.



Figure 2: *Left: Water tank with lifting body. Right: Pneumatic control valve (Samson, Type 3241, Series 250) for controlling the air flow to the cylinder.*

in this sense, but to design and implement a proper position control system for the selected configuration, assuming that the principal results are transferable to a different configuration.

3 Mathematical Modeling

3.1 Variables and parameters

Variables and parameters with values are given below.

- y [m] is position of lift body. Position is zero at the water surface, and positive direction is downwards.
- f_{lift} [N] is buoyancy lifting force.
- f_h [N] is hydrodynamic drag or damping (friction) force acting on the lifting body from the water.
- f_g [N] is gravity force on the system.
- f_d [N] is any independent environmental force in addition to the forces defined above.
- m_{tot} [kg] is total (resulting) mass to be lifted by the lifting body.
- m_b [kg] is mass of the ballast water.
- m_{air} [kg] is mass of air in air pocket
- $m_{add} = 2.0$ [kg] is added (virtual) mass related to the forced motion of water as the lifting body moves.
- $m_{cyl} = 6.0$ [kg] is mass of the body.

- $m_{cyl_{lift}} = 2.0$ [kg] is mass of the water corresponding to volume taken up by the lifting body in the water (causing buoyance force).
- $m_l = 5.0$ [kg] (default value) is mass of load attached to the lifting body.
- V_b [m³] is volume of ballast.
- V_{cyl} [m³] is volume of lifting body.
- V_a [m³] is volume of air pocket in lifting body.
- A_{cyl} [m²] is cross-sectional area of lifting body.
- A_p [m²] is a cross-sectional area parameter used to calculate f_d .
- $L_{cyl} = 0.6$ [m] is length of lifting body.
- $d = 0.2$ [m] is lifting body diameter.
- L_a [m] is height of air pocket in lifting body.
- ρ_a [kg/m³] is density of air in air pocket (at the present depth).
- $\rho_{atm} = 1.2$ [kg/m³] is density of air at the surface (atmospheric pressure).
- $\rho_w = 1025$ [kg/m³] is water density.
- F_{in} [kg/s] is air mass flow into lifting body from the compressor, through the control valve.
- F_{out} [kg/s] is air mass flow out from lifting body through outlet valve.
- $K_h = 0.004$ [N/(m/s)²] is hydrodynamic drag or damping (friction) force.
- $K_v = 250$ [kg/s] is valve constant of inlet (control) valve.
- $K_{v_{out}}$ is valve constant of outlet (manual) valve.
- p_{atm} [Pa] is atmospheric air pressure.
- p_s [Pa] is the pressure of the air out from air compressor (air supply).
- Δp_{in} [Pa] is pressure drop across air inlet valve.
- Δp_{out} [Pa] is pressure drop across air outlet valve.
- $\Delta p_{atm} = 101000$ [Pa] is atmospheric pressure.
- u [V or %] is valve control signal.
- $g = 9.81$ [m/s²] is gravity constant.

The parameters K_h and $K_{v_{out}}$ were adjusted manually in a simulator until the simulated position showed the similar response as the measured position. The parameter m_{add} was given an assumably reasonable value.

3.2 Mathematical modelling

A mathematical model describing the motion of the lifting body was developed from the following modeling principles:

- Equation of motion of the lifting body with load
- Mass balance of the air in the air pocket of the lifting body

Modeling details are given in the following sections.

3.2.1 Equation of motion

Applying Newton's Second Law gives

$$m_{tot}\ddot{y} = -f_{lift} - f_h + f_g + f_d \quad (1)$$

m_{tot} in eq. (1) is described in detail below.

$$m_{tot} = m_b + m_{add} + m_{cyl} + m_l \quad (2)$$

where

$$m_b = \rho_w V_b \quad (3)$$

$$V_b = V_{cyl} - V_a \quad (4)$$

$$V_a = \frac{m_a}{\rho_a} \quad (5)$$

m_a is given by the mass balance of the air in the lifting body, cf. eq. (20) below. Now, eq. (1) becomes

$$m_{tot} = \rho_w \left(V_{cyl} - \frac{m_a}{\rho_a} \right) + m_{add} + m_{cyl} + m_l \quad (6)$$

where

$$V_{cyl} = A_{cyl} L_{cyl} \quad (7)$$

In eq. (6), ρ_a is a function of depth y given by the Gas Law:

$$\rho_a(y) = \frac{p(y)}{RT} \quad (8)$$

R is the specific gas constant, and T is the temperature. Assuming constant T , eq. (8) gives

$$\frac{\rho_a(y)}{p_a(y)} = \frac{\rho_a(0)}{p_a(0)} \equiv \frac{\rho_{atm}}{p_{atm}} \quad (9)$$

Here,

$$p_a(y) = p_{atm} + \rho_w g (L_a + y) \quad (10)$$

Now eq. (9) gives

$$\rho_a(y) = \rho_{atm} \frac{p_a(y)}{p_{atm}} \quad (11)$$

$$= \rho_{atm} \left[1 + \frac{\rho_w g (L_a + y)}{p_{atm}} \right] \quad (12)$$

Here we will make the assumption that

$$y \gg L_a \quad (13)$$

which will hold when the lifting body is at relatively large depths. This implies

$$\rho_a(y) \approx \rho_{a_{atm}} \left(1 + \frac{\rho_w g y}{p_{atm}} \right) \quad (14)$$

which is used in eq. (6). Each of the terms at the right side of eq. (1) are now described: In eq. (1), the lifting force f_{lift} is given by

$$f_{lift} = \rho_w V_a g \quad (15)$$

where

$$V_a = A_{cyl} L_a \quad (16)$$

$$A_{cyl} = \frac{\pi d^2}{4} \quad (17)$$

In eq. (1),

$$f_h = K_h \frac{dy}{dt} \left| \frac{dy}{dt} \right| \quad (18)$$

where K_h is adjusted manually during experiments.

In eq. (1),

$$f_g = (m_l + m_{cyl} - m_{cyl_{lift}}) g \quad (19)$$

where the term $m_{cyl_{lift}} g$ is the buoyance force due to the lifting body itself being submerge.

3.2.2 Mass balance of air in air pocket

The mass m_a of the air in the air pocket of the lifting body is varying. A mathematical model of m_a is given by the following *mass balance* of the air in the lifting body:

$$\dot{m}_a = F_{in} - F_{out} \quad (20)$$

The air mass inflow F_{in} and the air mass outflow F_{out} are described in detail in the following.

Modeling the air inflow

F_{in} is assumed to be a controllable (adjustable) input variable. In practice the specified F_{in} is obtained by manipulating the control signal u to the inlet control valve. u is a voltage in the range of [0 – 5V] which will be represented with a percent value in the range of [0 – 100%], with a linear relation between the ranges. F_{in} is ideally given by the valve equation:

$$F_{in} = K_v f_{v_1}(u) \sqrt{\frac{\Delta p_{in}}{p_{atm}}} \quad (21)$$

Δp_{in} is the pressure drop across the valve:

$$\Delta p_{in} = p_s - p_a \quad (22)$$

Typically,

$$p_s \gg p_a \quad (23)$$

Hence,

$$F_{in} \approx K_v f_{v_1}(u) \sqrt{\frac{p_s}{p_{atm}}} \quad (24)$$

$f_{v_1}(u)$ is the valve flow characteristic function normalized with values between 0 and 1. The valve used in this project has an equal percentage valve characteristic function. However, since we have an air flow meter installed in the rig, we have decided to model the valve with experimental relation between control signal and flow. Fig. 3 shows the air flow F_{in} [kg/s] as a function of the valve control signal u [%]. The supply pressure was $p_s = 1$ bar. The circles are experimental data, and the lines are just linear interpolations.

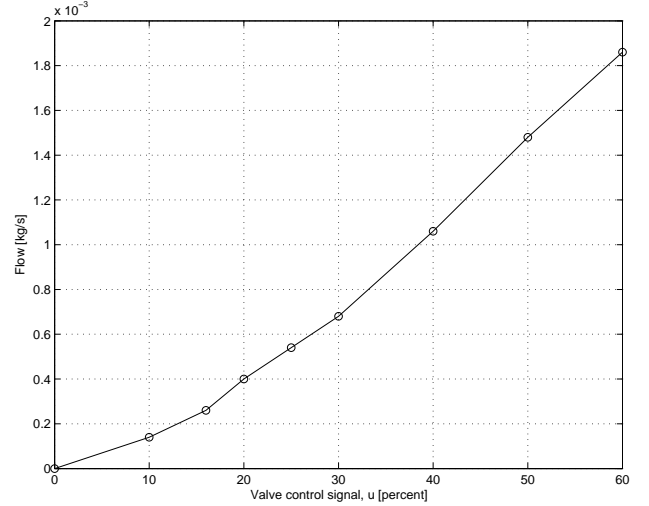


Figure 3: Air flow F_{in} [kg/s] as a function of the valve control signal.

In the project we *inverted* this valve function by using table-lookup based on linear interpolation between the tabular data. Hence, for any specified flow F_{in} , the table-lookup give the valve control signal u needed to obtain that flow. One convenient consequence of this is that the nonlinear valve can be represented by a linear valve with flow F_{in} as the control signal (or manipulating variable).

Modeling the air outflow

In eq. (20) the air mass outflow F_{out} through the valve, which has fixed opening, is modelled as

$$F_{out} = K_{v_{out}} \sqrt{\frac{|\Delta p_{out}|}{p_{atm}}} \cdot \text{sign}(\Delta p_{out}) \quad (25)$$

where Δp_{out} is the pressure drop across the valve. $\text{sign}(\Delta p_{out})$ is the sign function, which has value 1 if the argument (Δp_{out}) is positive, and -1 if the argument is negative. Δp_{out} is given by

$$\Delta p_{out} = \rho_w g L_a \quad (26)$$

4 Control System Design

4.1 Introduction

Several controllers were designed and tried out both on the simulator (based on the model described in Section 3) and the real system. The application of these controllers is described briefly below. All of the controllers worked on a simulator, but the only method which worked well and without problems on the physical system was cascade control, which also was enhanced with feedforward from estimated load force. (Section 4.2 describes details of the cascade control system.) One practical problem with the physical system is that the height of the water tank is relatively small as the height is about 1.5 m. Therefore, the lifting body with load mass easily reached physical constraints during experiments. Some of the other controllers might work well too under different physical conditions.

Certainly, it is beneficial if the control task is satisfactorily solved with a “standard” controller and this turned out to be the case in the present application (with cascade control).

Below is a short description of the application of the various controllers:

- Feedback linearization in combination with a Kalman filter which estimates the state variables.** The state variables are the lifting body position and velocity and the mass of air in the air pocket. Basically, with this controller, nonlinear (and linear) terms of the process model are cancelled out. The resulting process model is linear and simple (three integrators in series). A linear pole-placement controller for this model, augmented with an integrator in the controller to obtain integral control action, was designed. Although the controller worked excellently with the simulated system (with no model errors assumed), it was not able to stabilize the real system. This is probably because of too large sensitivity to model errors. Model errors are particularly apparent when the system reaches physical constraints, but this was not analyzed in detail. Furthermore, the sensitivity to an erroneous estimate of mass of air in the air pocket in the lifting body is probably large. The mass of air is closely related to the lifting force. Intuitively, an erroneously calculated
- lifting force may cause problems for stabilization of the body position.
- Linear feedback control with pole-design in combination with a Kalman filter which estimates the state variables.** This controller is based on a simplified process model being valid close to a certain operating point, which was selected as the “static” operating point, where the body is at rest (this is the most critical operating point regarding stability). This controller worked well on the simulated system, but not on the real system, probably because of the same reasons as for Feedback linearization (see above).
- PID controller with lead element in series with the PID controller.** Ordinary PID control could not stabilize the system at all due to the dynamic properties of the process which is roughly a triple integrator (three integrators in series), at least at a static operating point where damping hydrodynamic drag force (as modeled in this project) is zero. Position is integral of velocity which is integral of acceleration which is proportional to mass of air in air pocket, and this mass is roughly the integral of air inflow. A PID controller can stabilize a double integrator, but not a triple integrator because a phase lead of more than 90 degrees provided by the controller is needed, and the PID controller can not add more positive phase (lead action) than 90 degrees (ideally) to the control loop. Therefore, a lead-lag element with dominant lead action providing the additional phase lead was included in the controller, succeeding the PID controller. Actually, this controller was able to stabilize the real system, but it became unstable after small disturbances (perturbations), so it was concluded that the controller was not sufficiently robust. Also, the control signal was very noisy, due to the derivative action corresponding to the phase lead action. This noise makes the stabilization difficult.
- PID controller together with body acceleration feedback, where the acceleration was estimated with a state observer.** The purpose of acceleration feedback is similar to the lead-lag element mentioned above, i.e. to add sufficient positive phase in the control loop. The observer, which is designed from specified observer poles, estimates the lifting bag position, velocity, and acceleration using only position measurement. The experiences with this controller are similar with the experiences with PID controller with lead element. An alternative to estimating the acceleration is, of course, to measure it, but this was not

implemented in this project. (So, it is unclear if control using acceleration measurement works better than using acceleration estimate.)

- Cascade control, with feedforward from estimated load force.** The primary (master) loop of the cascade control system realizes position control, and the secondary (slave) loop realizes control of air height in the air pocket of the lifting body. Cascade control is “industry standard”, and it worked successfully without problems in this project. In the controller design, the clue is to identify the secondary process variable to be measured. The process model was a great help to this end as the model revealed that the state variables are lifting body position and velocity and mass of air in the air pocket. The primary controller is of course based on the lifting body position, and indirectly on velocity which is calculated by the derivative action of the PID controller. The secondary controller is based on the third process state variable - the air mass. (The cascade controller resembles state-variable feedback, which is known to be able to stabilize “any” process.) However, this mass is not measured directly. Instead, a closely related variable, namely the height of the air pocket, is measured. The cascade controller was enhanced with feedforward from estimated load force. Details about the cascade controller and is given in Section 4.2, and details about estimation of load force and feedforward from the force estimate is given in Sections 4.3 and 4.4, respectively.

4.2 Cascade control

4.2.1 Control system structure

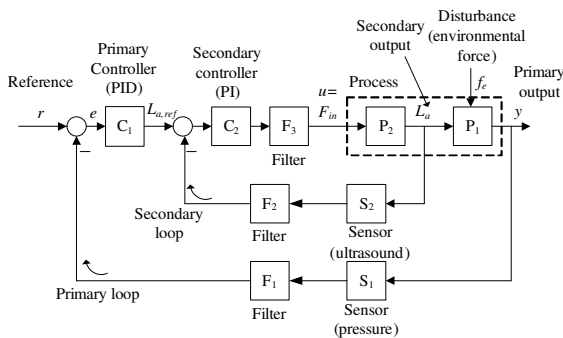


Figure 4: *Cascade control system.*

Fig. 4 shows a block diagram of the cascade control system based on feedback from measured underwater lifting body position y (this feedback is of course obligatory in a positional control system) and measured air

height L_a of the air pocket. Thus, two sensors were implemented.

Since a mathematical model exists, it was tempting to try feedback from estimated L_a as an alternative to feedback from measured L_a . However, this did not work well on the physical system as the control system became unstable. The system relies on accurate knowledge about L_a , and an estimate can not replace the measurement in this case.

One practical benefit of cascade control compared with more complex control structures is that the user can manipulate the amount of air (the secondary or internal process variable) directly during testing etc. This is done by setting the primary controller into manual mode, and manipulating the manual control signal of the primary controller which is the setpoint of the secondary controller.

4.2.2 Signal filters

Lowpass (time-constant) filters were implemented in LabVIEW to smooth the y measurement – this is filter F_1 in Fig. 4 – and the L_a measurement – this is filter F_2 in Fig. 4. Also, a lowpass filter was implemented to smooth the control signal, $u = F_{in}$ – this is filter F_3 in Fig. 4. In the experiments the F_1 filter was used with time-constant of 0.1 s because that measurement was somewhat noisy. The F_3 filter was used with a time-constant of 0.1 s to get a smooth control signal to the valve. The F_2 filter was actually not used (i.e. its time-constant was set to zero) because the measurement of L_a contained little noise.

4.2.3 Controller tuning

The master PID controller and the slave PI controller were both tuned using the mathematical model of the system.⁴ Both controllers were tuned from specifications of the response-time (time-constants) of the respective control loops using Skogestad’s method of PID controller tuning Skogestad (2003).

Tuning of the primary controller (PID) The primary control variable is L_a , which is used as the setpoint of the secondary loop. It is assumed that the dynamics of the secondary loop which controls L_a is so fast relative to the dynamics of the primary loop that the setpoint of L_a is obtained approximately immediately. Thus, the process model used as the basis for tuning the primary loop is as follows:

$$m_{tot}\ddot{y} = -f_{lift} - f_h + f_g + f_d \quad (27)$$

⁴And there was actually no need to retune the controllers when the system was online (i.e. connected to the physical system)!

where

$$m_{tot} = m_b + m_{add} + m_{cyl} + m_l \quad (28)$$

$$m_b = \rho_w (V_{cyl} - A_{cyl} L_a) \quad (29)$$

$$f_{lift} = \rho_w A_{cyl} g L_a \quad (30)$$

$$f_h = K_h \dot{y} \quad (31)$$

$$f_g = (m_l + m_{cyl} - m_{cyl_{lift}}) g \quad (32)$$

To tune the primary PID controller, a transfer function model with L_a as input and y as output is useful. This model can be derived by rigorous linearization of a nonlinear state-space model corresponding to eq. (27). Alternatively, it can be derived as follows: Since the hydrodynamic damping force f_h given by eq. (31) has its minimum value of zero when the speed \dot{y} is zero, the most critical operating point – which should be used for controller tuning – is at $\dot{y} = 0$. From the model eqs. (27) – (32), the relation between L_a and y at this operating point is

$$m_{tot} \ddot{y}(t) = -\rho_w A_{cyl} g L_a(t) \quad (33)$$

From eq. (33) the transfer function from L_a to y is found as

$$\frac{y(s)}{L_a(s)} = P_1(s) = \frac{K_1}{s^2} \quad (34)$$

where the gain K_1 is

$$K_1 = -\frac{\rho_w A_{cyl} g}{m_{tot}} \quad (35)$$

$P_1(s)$, which is a “double integrator”, can be controlled with a PID controller. To tune the controller parameters, Skogestad’s method, also denoted the SIMC method⁵, Skogestad (2003) is used:

$$K_p = \frac{1}{4K_1 (T_{C_1})^2} \quad (36)$$

$$T_i = 4T_{C_1} \quad (37)$$

$$T_d = 4T_{C_1} \quad (38)$$

where T_{C_1} is the user-specified time-constant of the control system (the primary loop). In general for the double-integrator being tuned with Skogestad’s formulas, it turns out that the actual (simulated) time-constant is about three times larger than the specified T_{C_1} . But since T_{C_1} has to be manually adjusted on the real system, this inaccuracy is not important. T_{C_1} is the only tuning parameter of the PID controller. T_{C_1} can be selected by trial-and-error on a simulator, and should be further tuned on the real control system. It was found that

$$T_{C_1} = 2.5 \text{ s} \quad (39)$$

was a good value on the simulator and on the real system.

Skogestad’s tuning rule assumes a *serial* PID controller function, which has the following transfer function (assuming u is the control signal and e is the control error):

$$u(s) = L_a(s) = K_{p_s} \frac{(T_{i_s} s + 1)(T_{d_s} s + 1)}{T_{i_s} s} e(s) \quad (40)$$

where K_{p_s} , T_{i_s} , and T_{d_s} are the controller parameters given by eqs. (36) – (38). However, the PID controller used in this project actually implements a *parallel* PID controller, which has the following transfer function:

$$u(s) = L_a(s) = \left[K_{p_p} + \frac{K_{p_p}}{T_{i_p} s} + K_{p_p} T_{d_p} s \right] e(s) \quad (41)$$

To transform the serial PID settings to parallel PID settings, we apply the following serial-to-parallel transformations Skogestad (2003):

$$\begin{aligned} K_{p_p} &= K_{p_s} \left(1 + \frac{T_{d_s}}{T_{i_s}} \right) = \frac{1}{4K (T_{C_1})^2} \left(1 + \frac{4T_{C_1}}{4T_{C_1}} \right) \\ &= \frac{1}{2K_1 (T_{C_1})^2} = K_p \end{aligned} \quad (42)$$

$$\begin{aligned} T_{i_p} &= T_{i_s} \left(1 + \frac{T_{d_s}}{T_{i_s}} \right) = 4T_{C_1} \left(1 + \frac{4T_{C_1}}{4T_{C_1}} \right) \\ &= 8T_{C_1} = T_i \end{aligned} \quad (43)$$

$$\begin{aligned} T_{d_p} &= T_{d_s} \frac{1}{1 + \frac{T_{d_s}}{T_{i_s}}} = 4T_{C_1} \frac{1}{1 + \frac{4T_{C_1}}{4T_{C_1}}} \\ &= 2T_{C_1} = T_d \end{aligned} \quad (44)$$

Tuning of the secondary controller (PI) The secondary control loop controls L_a . The setpoint is $L_{a_{ref}}$ which is equal to the output of the primary controller. The control variable calculated by the secondary controller is F_{in} , which is applied to the control valve. The mathematical model of the process that the secondary controller controls, is given by eq. (20) which is repeated here:

$$\dot{m}_a = F_{in} - F_{out} \quad (45)$$

The relation between L_a and m_a is given by

$$m_a = \rho_a V_a = \rho_a A_{cyl} L_a \quad (46)$$

Above, the air density ρ_a is given by eq. (14). Because y is known at any instant of time, ρ_a can be

⁵SIMC = Simple Internal Model Control

calculated by eq. (14) and is therefore known. Furthermore, we can assume that the outflow F_{out} is a disturbance. Then, the model of the process that the secondary controller controls, is

$$\rho_a A_{cyl} \dot{L}_a = F_{in} - F_{out} \quad (47)$$

The transfer function from control variable F_{in} to process output variable L_a becomes

$$\frac{L_a(s)}{F_{in}(s)} = P_2(s) = \frac{K_2}{s} \quad (48)$$

which is an integrator. The gain K_2 is

$$K_2 = \frac{1}{\rho_a A_{cyl}} = \frac{1}{\rho_{a_{atm}} \left(1 + \frac{\rho_w g y}{\rho_{a_{atm}}}\right) A_{cyl}} \quad (49)$$

To tune the secondary controller for a process given by eq. (48), we use Skogestad's method Skogestad (2003):

$$K_p = \frac{1}{K_2 T_{C_2}} \quad (50)$$

$$T_i = 1.5 T_{C_2} \quad (51)$$

$$T_d = 0 \quad (52)$$

where T_{C_2} is the user-specified time-constant of the secondary control system. It was found that

$$T_{C_2} = 1.0 \text{ s} \quad (53)$$

is a good value on the simulator and on the real system.

4.3 Estimation of load force

An estimator of the load force was implemented. The force estimate was used in feedforward control as explained in Section 4.4. The estimator was designed as an *observer* with specified dynamics, see Goodwin et al. (2001). The design is described in the following.

The mathematical model which is the basis of the estimator is

$$m_{tot} \ddot{y} = \underbrace{-\rho_w A_{cyl} g L_a}_{f_{i ft}} - \underbrace{K_h \dot{y} |\dot{y}|}_{f_h} + f_e \quad (54)$$

where f_e is the environmental or disturbance force to be estimated. f_e will actually represent any force that is not included in the model, or that is not modelled correctly. It is assumed that f_e is unknown without any information about its variation. An appropriate model which describes f_e is therefore

$$\dot{f}_e = 0 \quad (55)$$

It is assumed that all variables and parameters except f_e are known. In particular, L_a is known from its measurement. In applications where the m_{tot} varies substantially, it may be necessary to estimate it, but this has not been done in the present project.

f_e will be estimated with an observer, which is a state-estimator designed from pole (or eigenvalues) specifications. (An observer has the same structure as the Kalman filter.) It is the difference between the measured and the estimated position, $e = y_{meas} - y_{est}$, that updated the estimates, including the estimate of f_e . To design the observer, we need to write the model in eqs. (54) - (55) as a state-space model (v is velocity):

$$\dot{y} = v \quad (56)$$

$$\dot{v} = -\frac{\rho_w A_{cyl} g}{m_{tot}} L_a - \frac{K_h \dot{y} |\dot{y}|}{m_{tot}} + \frac{1}{m_{tot}} f_e \quad (57)$$

$$\dot{f}_e = 0 \quad (58)$$

The observer is given by

$$\dot{y}_{est} = v_{est} + K_1 e \quad (59)$$

$$\begin{aligned} \dot{v}_{est} = & -\frac{\rho_w A_{cyl} g}{m_{tot}} L_a - \frac{K_h v_{est} |v_{est}|}{m_{tot}} \\ & + \frac{1}{m_{tot}} f_{e_{est}} + K_2 e \end{aligned} \quad (60)$$

$$\dot{f}_{e_{est}} = K_3 e \quad (61)$$

K_1 , K_2 and K_3 are observer gains. Fig. 5 shows a block diagram of the observer.

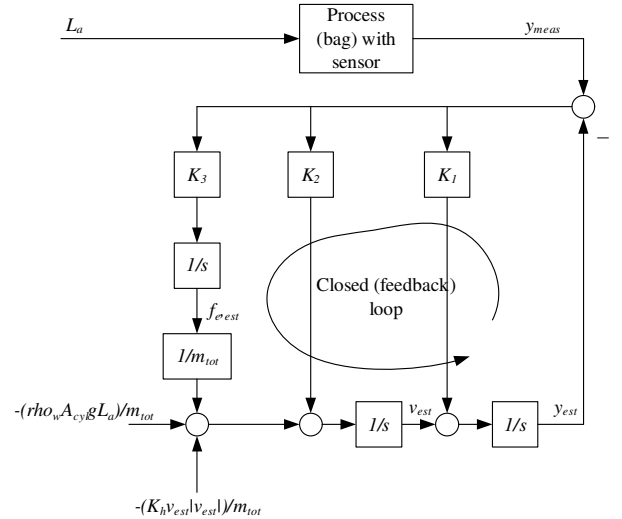


Figure 5: The observer which is used to estimate the environmental force f_e . ($1/s$ is the transfer function of an integrator.)

The observer gains are calculated from specified dynamics in terms of poles of the closed (feedback) loop

of the observer. It can be shown that the characteristic polynomial of this loop is

$$c_{obs}(s) = s^3 + K_1 s^2 + K_2 s + \frac{K_3}{m_{tot}} \quad (62)$$

The observer poles were selected as Butterworth poles [Franklin and Powell \(1980\)](#). Butterworth poles give a step response with a slight overshoot and good damping. (Of course, other poles configurations are possible.) Butterworth poles of a normalized third order system correspond to the following characteristic polynomial:

$$c_{butter_{norm}}(s) = c_3 s^3 + c_2 s^2 + c_1 s + c_0 \quad (63)$$

where

$$c_3 = 1; c_2 = 2; c_1 = 2; c_0 = 1 \quad (64)$$

If we specify that the system has response-time⁶ T_r approximately equal to $3T$, the Butterworth polynomial becomes [Haugen \(2010\)](#)

$$c_{butter_1}(s) = c_3 (Ts)^3 + c_2 (Ts)^2 + c_1 (Ts) + c_0 \quad (65)$$

which has the same roots as

$$c_{butter}(s) = s^3 + \frac{c_2}{c_3} \frac{1}{T} s^2 + \frac{c_1}{c_3} \frac{1}{T^2} s + \frac{c_0}{c_3} \frac{1}{T^3} \quad (66)$$

Comparing $c_{obs}(s)$ and $c_{butter}(s)$ gives the following formulas for the observer gains:

$$K_1 = \frac{c_2}{c_3} \frac{1}{T} \quad (67)$$

$$K_2 = \frac{c_1}{c_3} \frac{1}{T^2} \quad (68)$$

$$K_3 = \frac{c_0}{c_3} \frac{1}{T^3} m_{tot} \quad (69)$$

The algorithm of the observer ready for programming is derived by discretizing the observer formulas in eqs. (59) – (61) using Forward differentiation approximation. The resulting observer formulas are as follows:

$$y_{est}(t_{k+1}) = y_{est}(t_k) + h [v_{est}(t_k) + K_1 e(t_k)] \quad (70)$$

$$v_{est}(t_{k+1}) = v_{est}(t_k)$$

$$+ h \left[\frac{-\frac{\rho_w A_{cyl} g}{m_{tot}} J_a(t_k) - \frac{K_h v_{est}(t_k) |v_{est}(t_k)|}{m_{tot}}}{+\frac{1}{m_{tot}} f_{e_{est}}(t_k) + K_2 e(t_k)} \right] \quad (71)$$

$$f_{e_{est}}(t_{k+1}) = f_{e_{est}}(t_k) + h K_3 e(t_k) \quad (72)$$

where

$$e(t_k) = y_{meas}(t_k) - y_{est}(t_k) \quad (73)$$

h is the time-step (0.02 s). The observer response-time T_r was set to 0.8 s.

⁶Here, response-time means time to reach 63% of steady-state value, similar to the definition of time-constant for first order systems.

4.4 Feedforward from estimated load force

From eq. (54) we get the following feedforward controller function:

$$L_{a_{ff}} = \frac{1}{\rho_w A_{cyl} g} f_{e_{est}} \quad (74)$$

which is added to the control output from the primary controller, $L_{a_{prim}}$, so that the total reference to the secondary controller becomes

$$L_{a_{ref}} = L_{a_{prim}} + L_{a_{ff}} \quad (75)$$

Fig. 6 shows the structure of the cascade control system including estimator (observer) and feedforward model.

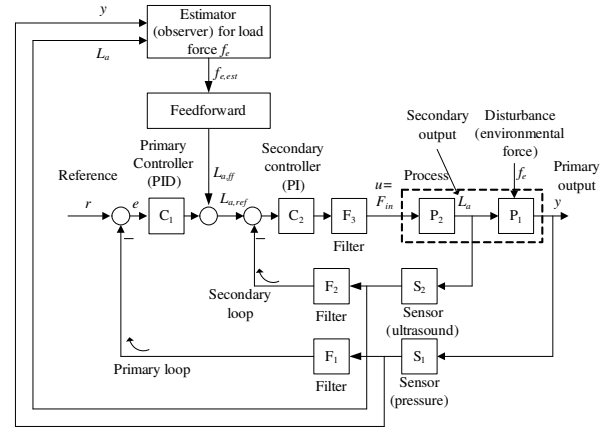


Figure 6: The structure of the cascade control system including estimator (observer) and feedforward model.

5 Experimental Results

5.1 Introduction

Experiments were run with cascade control with feedforward from estimated load force. Fig. 7 shows the available parameters of the control system. The experiments reported in the next sections are about

- *position reference tracking*, where the reference was changed as a ramp, and
- *disturbance (load) compensation*, where the load was suddenly changes by adding a mass of 0.5 kg. The reference was then constant at 0.3 m.

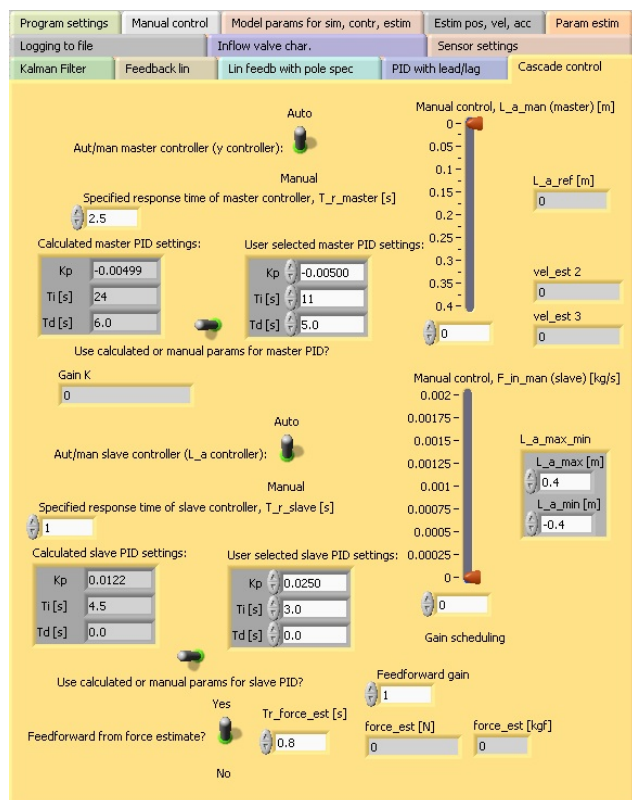


Figure 7: Controller settings of the cascade control system in LabVIEW.

5.2 Position reference tracking

Fig. 8 shows the lifting body position reference (set-point) r and measurement y , together with air body height reference $L_{a,ref}$ and measurement L_a . Also the control signal F_{in} applied to the control valve is shown. The position reference was initially 0.3 m, and then changed as a ramp downwards with slope 0.005 m/s, and then upwards with the same slope. When the reference was constant (before the ramping started), the maximum position control error was approximately 0.02 m, due to various sources of noise in the system (the error is zero in an ideal, noise-free system). During the ramping, the maximum steady-state error was approximately 0.03 m. In an earlier experiment, cascade control *without* feedforward from estimated force was run. The control error was substantially larger than with feedforward. (No plots are shown here.) The superior control at using feedforward from estimated load was somewhat surprising, because the load was not changed during these experiments. It seems that the load estimate encapsulates the effects of various sources causing the control error to become different from zero.

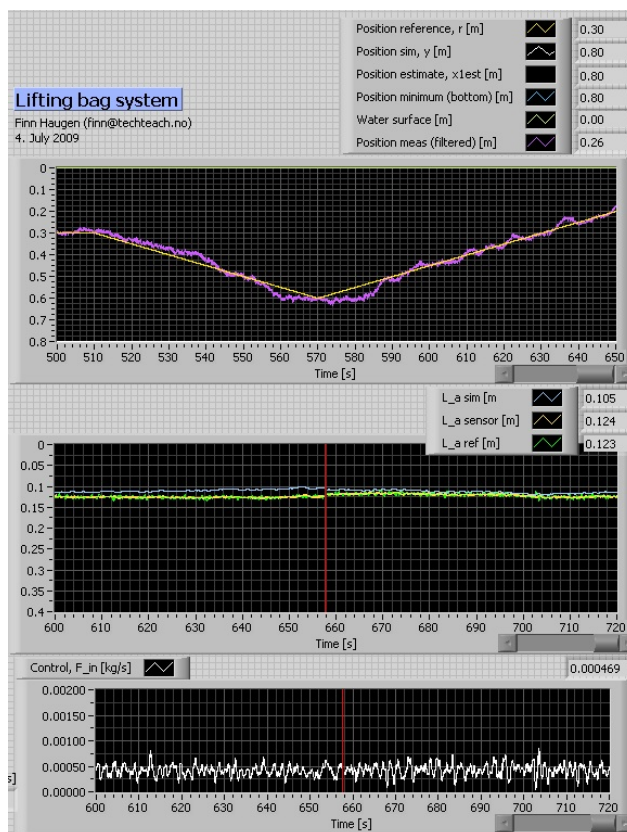


Figure 8: Cascade control with feedforward from estimated load force: Ramped changes of the position reference.

5.3 Disturbance (load) compensation

Fig. 9 shows the responses as a mass load of 0.5 kg was suddenly added to the body at time 360 s, and then suddenly removed from the body at time 410 s. The maximum response in the lifting body after the load was added was 0.08 m, while the maximum response after the load was removed was 0.12 m – an average of 0.10 m. It took about 15 sec until the position was back at the reference.

In another similar experiment *without* feedforward the maximum response in the lifting body was 0.23 m, and position was back at the reference after 250 sec. Thus, disturbance compensation was substantially improved with feedforward.

6 Conclusions

A mathematical model has been derived from physical principles. The state variables are lifting body position and speed, and mass of air in the air pocket of the lifting body. Successful stabilizing position control of the lifting body was obtained with cascade control.



Figure 9: Cascade control with feedforward from estimated load force: A load mass of 0.5 kg was added at time 360 s, and then removed at time 410 s.

The cascade control system is based on feedback from both measured lifting body position and measured air height in the lifting body. Thus, two sensors were implemented. Both the master PID controller and the slave PI controller were tuned using Skogestad's model-based PID tuning rules. The only specifications for Skogestad's tuning is the time-constant of the control loop. The tunings were tried on the simulator before being applied to the real system, and there was hardly any need to retune the controllers. Feedforward from estimated load force was implemented with the cascade control system. At constant position reference the maximum control error was 0.02 m. At ramping position reference of slope 0.005 m/s the maximum control error was 0.03 m. At constant reference but with a suddenly added load mass of 0.5 kg the maximum control error was 0.10 m in average (0.08 m after the load was added, and 0.12 m after the load was removed), and the position was back at the reference after 15 s. Experiments with cascade control but without feedforward gave a control error that was substantially (many times) larger, both at changing reference and at changing disturbance (load mass).

Acknowledgments

I want to express my thanks to Eivind Fjelddalen at Telemark University College who constructed the test rig, and to Claus Christian Apneseth at Miko Marine who initiated the project and gave valuable inputs and comments as the project evolved.

References

- Fossen, T. *Guidance and Control of Ocean Vehicles*. John Wiley & Sons Ltd., 1994.
- Franklin, G. and Powell, J. *Digital Control of Dynamic Systems*. Addison-Wesley, 1980.
- Goodwin, C., Graebe, S., and Salgado, M. *Control System Design*. Prentice-Hall, 2001.
- Haugen, F. *Advanced Dynamics and Control*. TeachTech, 2010.
- Otterblad, S. and Doovertie, R. Lifting body for diving. 1985. US Patent number 4,498,408.
- Skogestad, S. Simple analytic rules for model reduction and pid controller tuning. *Journal of Process Control*, 2003. 13(4):291–309. doi:10.1016/S0959-1524(02)00062-8.